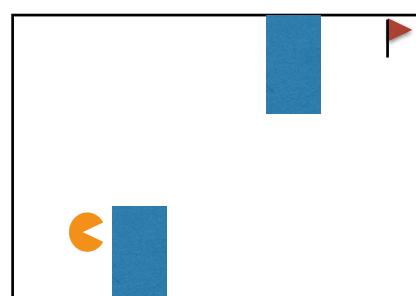


Recherche de plus court chemin



Comment atteindre le drapeau?

- Méthodes non informées

- Largeur d'abord
- Profondeur d'abord
- Uniform-cost (~Dijkstra)

- Méthodes informées (heuristique)

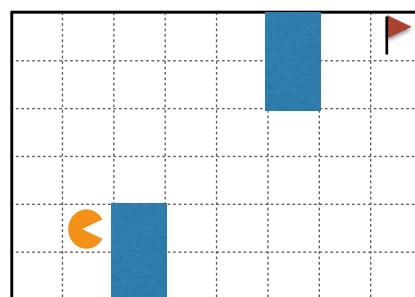
- Best-first (nécessite une heuristique pour évaluer les déplacements)
- Algorithme A*
- Améliorations de A* (D*, IDA*, Fringe, FSA*, GAA*, LPA*, SMA*, Jump point search, Theta*, ..., UCT)

Recherche non-informée

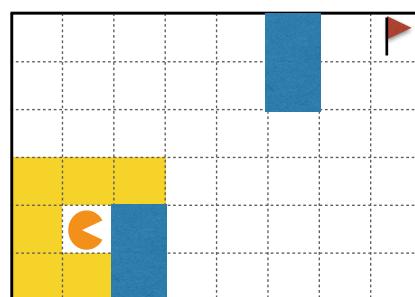
on ne dispose d'aucune information sur l'état final

Recherche en largeur d'abord

hypothèse: direction dans les 8 directions possibles

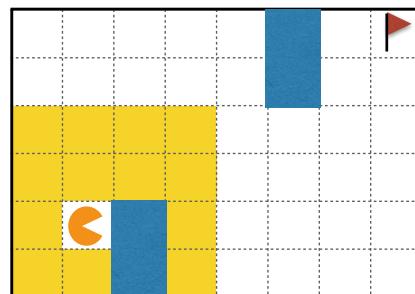


Recherche en largeur d'abord



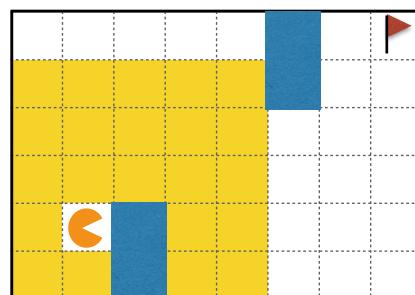
explore toutes les cases autour de la case de départ

Recherche en largeur d'abord



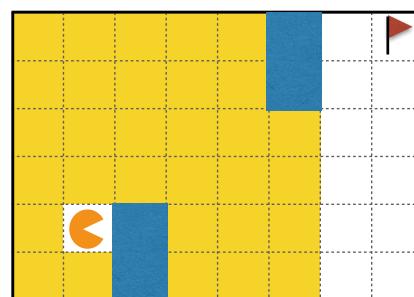
pour toutes les cases explorées, explore les cases voisines

Recherche en largeur d'abord

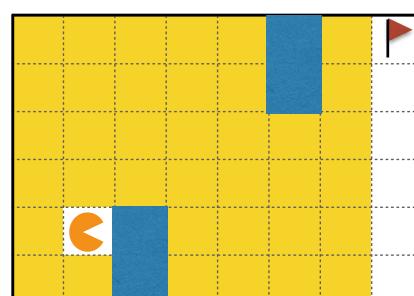


etc.

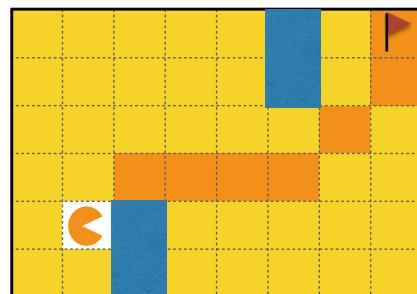
Recherche en largeur d'abord



Recherche en largeur d'abord



Recherche en largeur d'abord

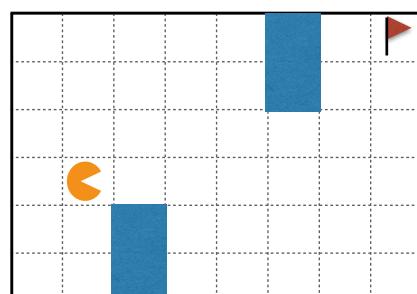


Exemple dans le pire cas
(mais cela dépend du problème)

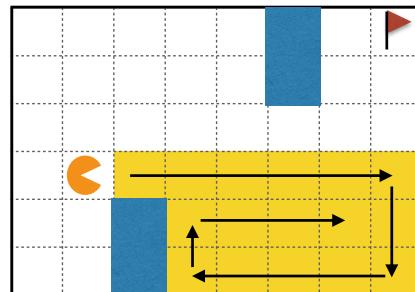
succès!

Recherche en profondeur d'abord

hypothèse: direction dans les 8 directions possibles

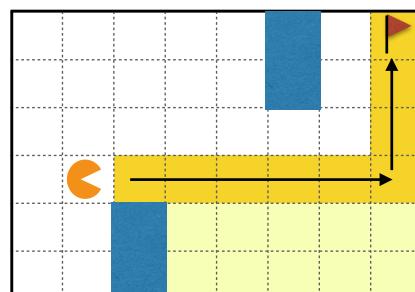


Recherche en profondeur d'abord



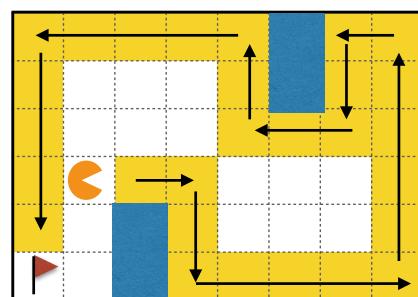
explore dans une direction choisie au hasard jusqu'à ne plus pouvoir se déplacer sans repasser sur une case déjà visitée

Recherche en profondeur d'abord



reprend l'exploration à partir de la dernière case avec une voisine pas encore explorée

succès!



Pas très optimal...

Uniform-cost

Ici les nombres désignent la **distance mesurée** depuis le point de départ de l'agent, notée $g(n)$

5	4	5	6	7	8	9	10
4	3	4	5	6	7	8	9
3	2					9	10
2	1	2	3	4			11
1	●	1	2	3			
2	1	2	3	4			

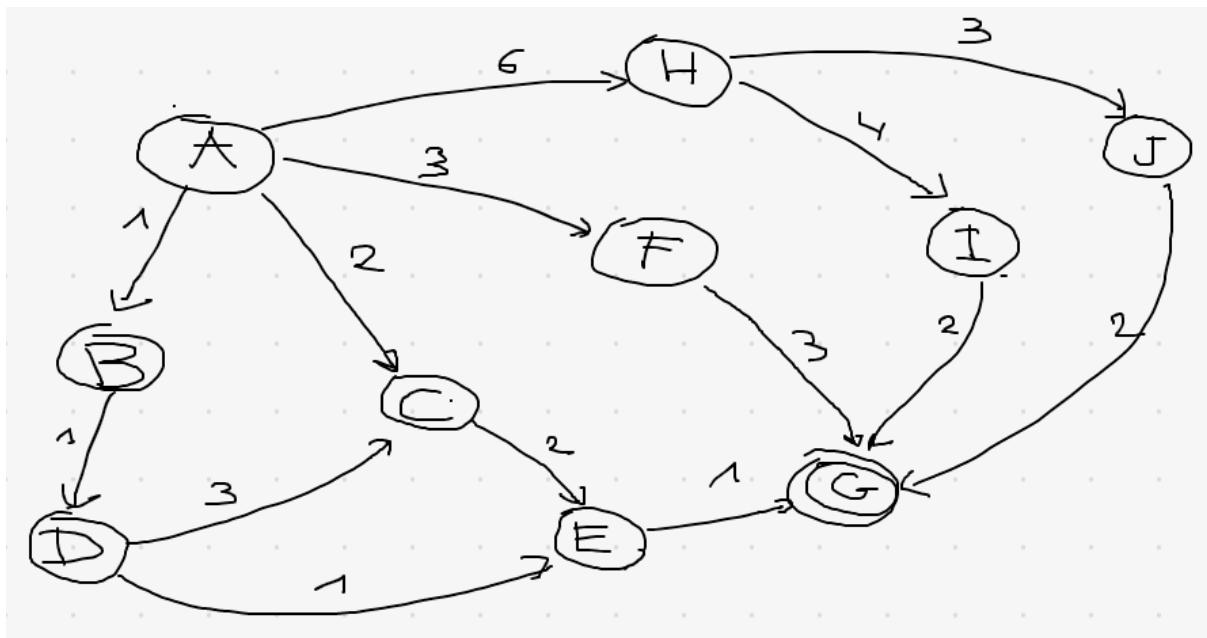
terminaison anticipé:
on a trouvé la cible!

(Dijkstra avec arrêt prématué)

Profondeur d'abord

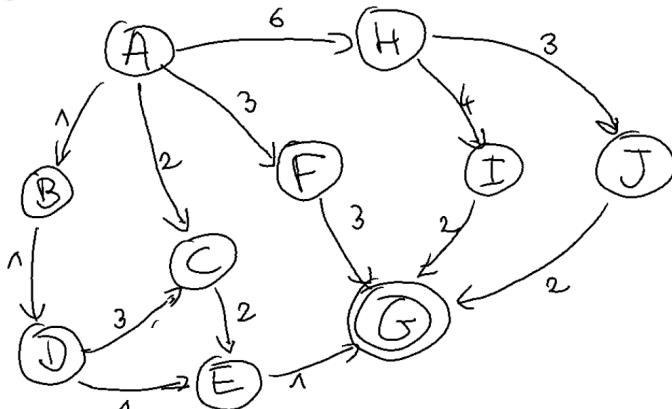
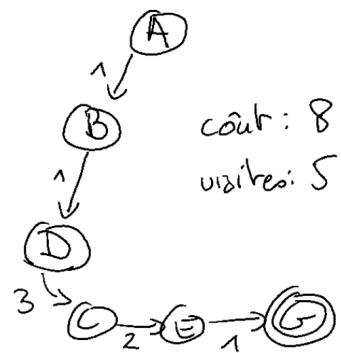
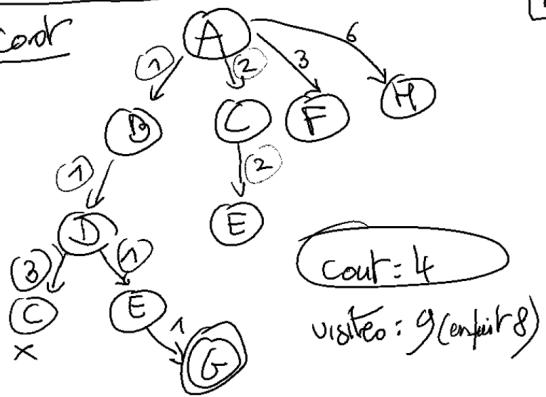
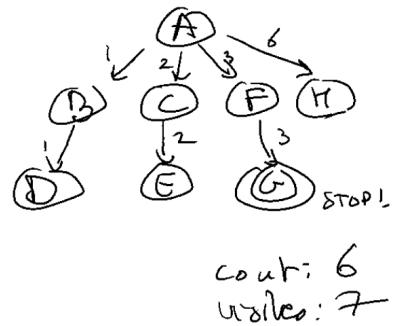
Largeur d'abord

Uniform cost



Hypothèse: ordre alphabétique (ex.: C vient avant E)

O

Profondeur d'abordRq: ordre alphabétique (d'où C auant à partir de D)Uniforme costLargem d'abord

Recherche informée

on dispose d'une heuristique pour estimer la distance à l'état final

Recherche avec une méthode réactive “naïve”

hypothèse: on dispose d'une heuristique qui donne une information sur la distance à la cible

7	6	5	4	3		1	🚩
8	7	6	5	4		2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	pacman		8	7	6	5	4
12	11		9	8	7	6	5

exemple: distance de manhattan à l'état cible

Recherche avec une méthode réactive “naïve”

7	6	5	4	3		1	🚩
8	7	6	5	4		2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	pacman		8	7	6	5	4
12	11		9	8	7	6	5

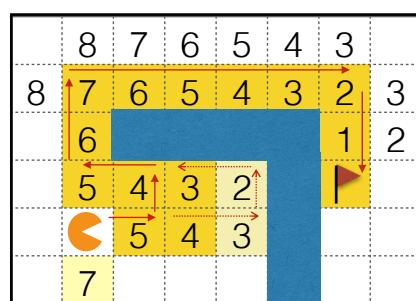
cas 1: déplacement NSEW
avec préférence déplacement horizontal
succès!

Recherche avec une méthode réactive “naïve” (2)

7	6	5	4	3		1	红旗
8	7	6	5	4		2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	●				8	7	6
12	11				5	4	

cas 2: déplacement NSEW
avec préférence déplacement vertical
bloqué!

Algorithme glouton



exploration dirigée par l'estimation de distance
sans repasser par les mêmes états

Trouve la cible... sans garantie de plus court chemin

A*

Ici les nombres désignent la **distance mesurée** depuis le point de départ de l'agent, notée $g(n)$

	4	5	6	7	8	9	
4	3	4	5	6	7	8	9
3	2				9	10	
2	1	2	3	4			🚩
1	💡	1	2	3			
	1	2	3	4			

Ici les nombres désignent la **somme** des deux distances, soit $g(n) + h(n)$

	12	12	12	12	12	12	
12	10	10	10	10	10	10	12
10	8						12
8	6	6	6	6			
8	6	6	6				
	8	8	8	8			

Ici les nombres désignent la **distance estimée** entre la case et le point d'arrivée, notée $h(n)$

	8	7	6	5	4	3	
8	7	6	5	4	3	2	3
7	6					1	2
6	5	4	3	2			
7	5	4	3				
	7	6	5	4			

score d'un état:
 $f(n) = g(n) + h(n)$

compromis entre le coût depuis le départ, et le coût estimé jusqu'à l'état cible

déplacement NSWE

A*

score d'un état:
 $f(n) = g(n) + h(n)$

avec:

- $f(n)$ score du noeud n (que l'on espère faible)
- $g(n)$ coût du chemin depuis l'état initial jusqu'au noeud n
- $h(n)$ coût estimé du chemin depuis le noeud n jusqu'à l'état but

L'algorithme A* retourne la solution optimale sous deux conditions:

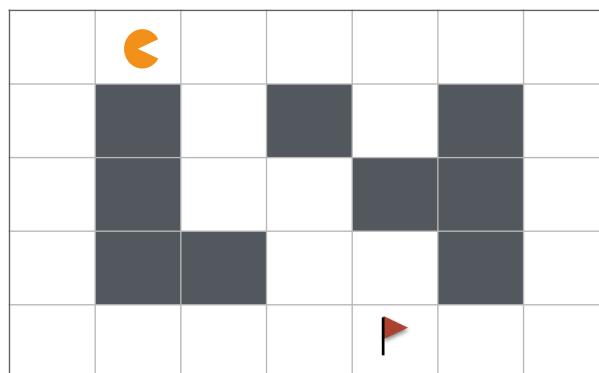
- **Admissibilité:** l'heuristique ne surestime jamais la distance à l'état but
- **Consistance:** les valeurs de $f(n)$ sont non-décroissantes sur les chemins depuis la racine

Déplacements

l'agent se déplace dans les quatre directions
 l'agent peut changer de direction aux embranchements
 l'agent ne peut pas faire de 180° ou revenir sur ses pas

Heuristique

distance jusqu'au but sans prendre en compte les murs



Déplacements

l'agent se déplace dans les quatre directions
 l'agent peut changer de direction aux embranchements
 l'agent ne peut pas faire de 180° ou revenir sur ses pas

Heuristique

distance jusqu'au but sans prendre en compte les murs



Algorithme Glouton

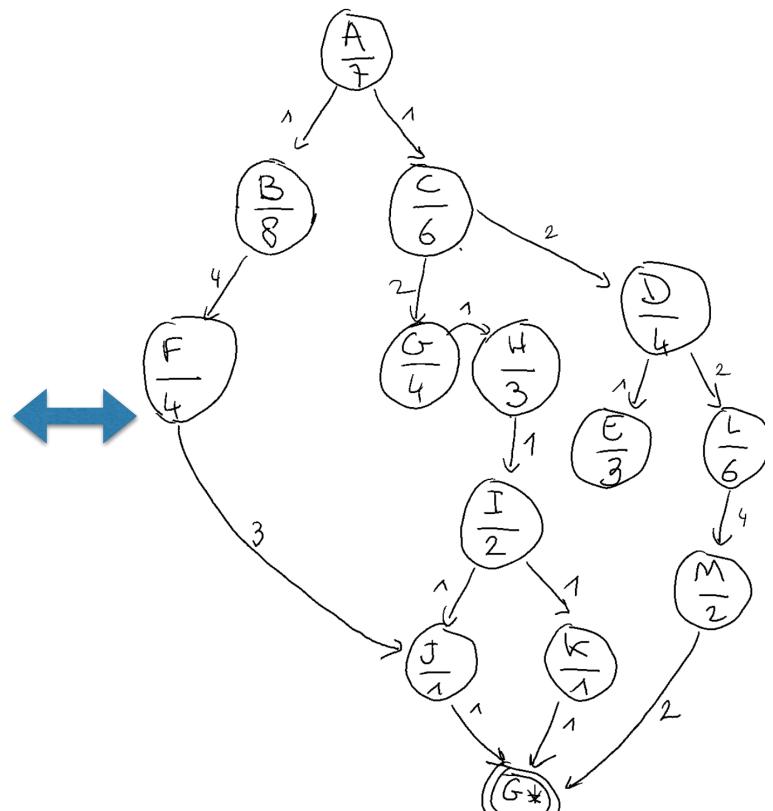
utilise uniquement le coût futur estimé $h(x)$

Algorithme A*

utilise le coût actuel $g(x)$ et le coût futur estimé $h(x)$

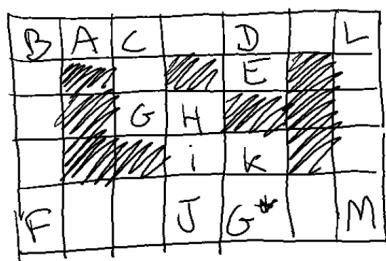
B	A	C	D	L
			E	
	G	H		
F		I	K	M

représentation sous forme de labyrinthe

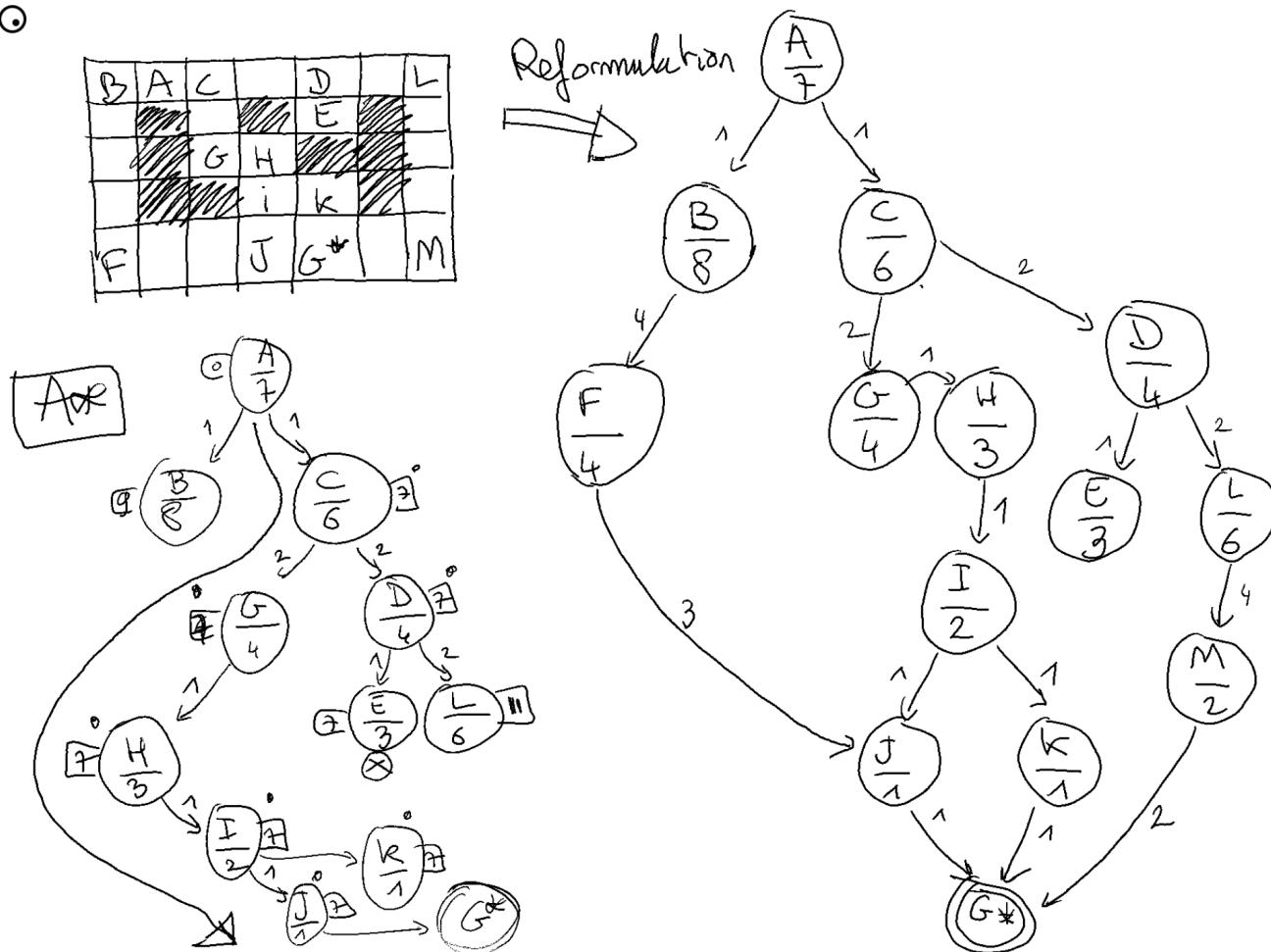


représentation sous forme de graphe

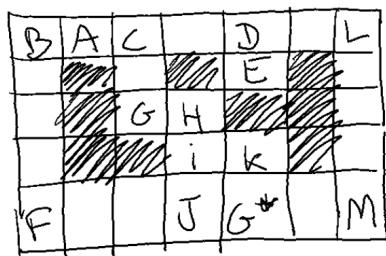
○



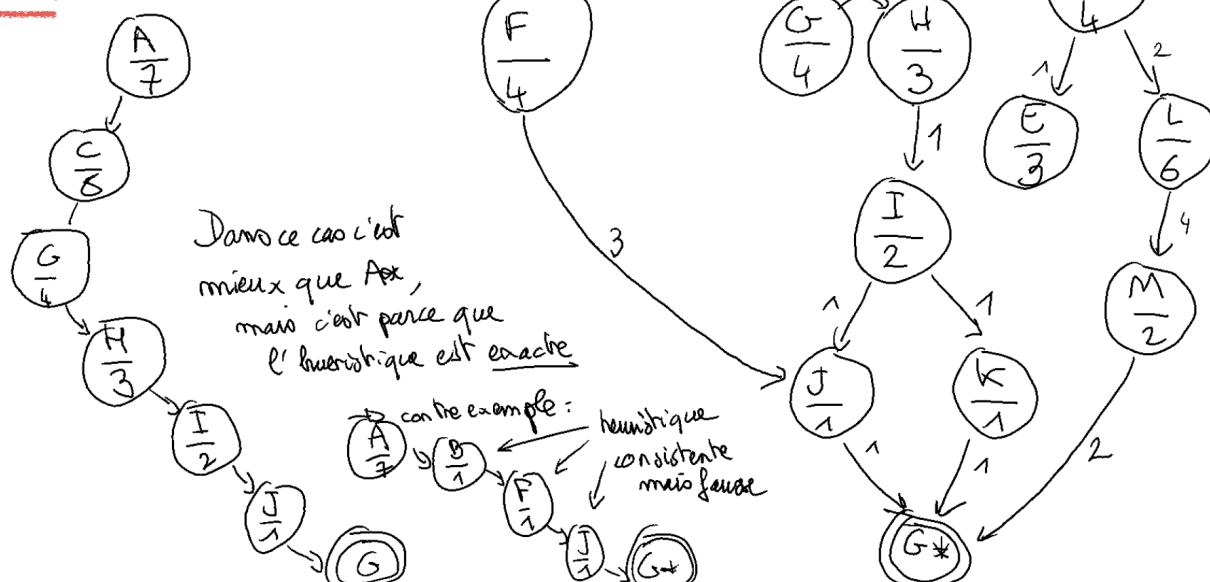
Reformulation



○



Reformulation

GLOUTON utilise uniquement l'heuristique)

Applications pratiques

34

Moodle, fichier “search_algorithm.ipynb”